



Universidade Federal de Alagoas (UFAL)
Campus Arapiraca



Programação Orientada a Objetos (POO)

11 - Introdução aos padrões de projeto

Alexandre de Andrade Barbosa

alexandre.barbosa@arapiraca.ufal.br

Objetivos

Objetivos

- Apresentar o conceito e motivar o uso de padrões de projeto

Padrões de projeto

- Criar um bom projeto de software é uma tarefa complicada
- Projeto O.O. reutilizável e flexível dificilmente é obtido na primeira especificação
- Projetistas experientes costumam realizar bons projetos, pois se baseiam em soluções anteriores
- Projetistas inexperientes tendem a iniciar cada novo projeto do zero

Padrões de projeto

Uma analogia...

- Autores de cinema, novela, teatro... reutilizam elementos básicos em seus roteiros



O vilão...O herói...A donzela...

Padrões de projeto

Uma analogia...

- Autores de cinema, novela, teatro... reutilizam elementos básicos em seus roteiros



O vilão...O herói...A donzela...

Padrões de projeto

- Muitos problemas que ocorrem em um projeto já devem ter sido resolvidos em projetos anteriores
- O problema é que dificilmente alguém irá lembrar como o problema foi resolvido
- Padrões de projeto consistem em registros destas experiências para o projeto de software

O que é um padrão de projeto?

“...cada padrão descreve um problema no nosso ambiente e o núcleo de sua solução...”

Christopher Alexander (Arquiteto)

O que é um padrão de projeto?

- Um padrão representa a solução **reutilizável** para um problema **recorrente** em um determinado contexto
- Os padrões são descritos utilizando quatro elementos essenciais:
 - nome - rótulo (sugestivo) utilizado para identificar o padrão
 - problema - descrição do tipo de problema e seu contexto
 - solução - descrição abstrata dos elementos do projeto utilizado para resolver o problema
 - consequências - lista das vantagens e desvantagens relacionadas ao uso do padrão
- O catalogo mais famoso de padrões é são os padrões GoF (*Gang of Four*)

Descrição dos padrões

- Os padrões de projeto GoF (*Gang of Four*) são descritos utilizando as seguintes descrições:
 - Nome e classificação
 - Intenção e objetivo
 - Sinônimos
 - Motivação
 - Aplicabilidade
 - Estrutura
 - Participantes
 - Colaborações
 - Conseqüências
 - Implementação
 - Exemplo de código
 - Usos conhecidos
 - Padrões relacionados

Descrição dos padrões

- Nome e classificação
 - O nome representa o padrão expressando seu objetivo de forma breve
 - A classificação descreve a finalidade e o escopo do padrão
- Intenção e objetivo
 - Declaração breve sobre o que o padrão faz, qual tipo de problema ele ataca, ...
- Sinônimos
 - Outros nomes pelos quais o padrão é conhecido

Descrição dos padrões

- Motivação
 - Descrição de um cenário ilustrativo do problema e de como o padrão soluciona este problema
- Aplicabilidade
 - Lista de situações onde o padrão pode ser aplicado com sucesso
- Estrutura
 - Diagrama expressando os elementos pertencentes ao padrão

Descrição dos padrões

- Participantes
 - Descrição das classes e/ou objetos que participam do padrão e quais são as responsabilidades associadas a cada um
- Colaborações
 - Descrição das colaborações entre os participantes
- Conseqüências
 - Lista de vantagens e desvantagens associadas ao uso do padrão

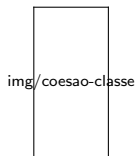
Descrição dos padrões

- Implementação
 - Sugestões e observações relacionadas aos cuidados que se deve ter na implementação do padrão
- Exemplo de código
 - Trechos de código implementando o padrão
- Usos conhecidos
 - Exemplos de sistemas que utilizam o padrão
- Padrões relacionados
 - Padrões que são relacionados ao padrão descrito

Descrição dos padrões

Coesão

- Em um sistema orientado a objetos uma classe deve ser responsável por um conjunto de responsabilidades
- No projeto orientado a objetos é desejável que as responsabilidades da classe sejam **coesas**
- **Coesão** corresponde ao quanto responsabilidades de uma classe estão relacionadas



Classe 'Conta' coesa, nenhum 'conteúdo' não relacionado a conta é tratado

Descrição dos padrões

Acoplamento

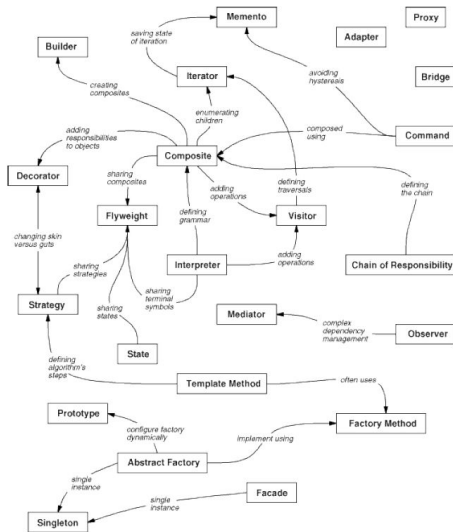
- Acoplamento = união ou ligação entre dois ou mais corpos, formando um único conjunto
- No contexto de orientação a objetos acoplar-se a uma classe significa depender de uma ou mais classes
- A dependência entre classes faz com que uma alteração em na classe 'principal' afete outras classes
- No projeto orientado a objetos é desejável que as classes possuam 'Acoplamento baixo'
- Um exemplo comum de alto acoplamento é o relacionamento de herança
- Acoplamento baixo - as classes tem pouca ou nenhuma dependência em relação as outras
- Acoplamento alto - as classes possuem dependência em relação as outras

Descrição dos padrões

Coesão e Acoplamento

- Alta coesão é uma propriedade desejável em sistemas orientados a objetos
- Um projeto com baixa coesão contém classes que são difíceis de entender, utilizar e manter
- Baixo acoplamento é uma propriedade desejável em sistemas orientados a objetos
- Um projeto com alto acoplamento contém classes que sofrem com mudanças constantes

Catálogo de padrões



Padrões e seus relacionamentos

Catálogo de padrões

23 padrões de projeto GoF, são eles:

Abstract Factory	Factory Method	Proxy
Adapter	Flyweight	Singleton
Bridge	Interpreter	State
Builder	Iterator	Strategy
Chain of Responsibility	Mediator	Template Method
Command	Memento	Visitor
Decorator	Observer	
Façade	Prototype	

Catálogo de padrões

- Os padrões de projeto variam em granularidade e em seu nível de abstração
- Dois critérios são adotados na classificação dos padrões de projeto GoF:
 - finalidade (criação, estrutura, comportamento)
 - escopo (classe, objeto)

Catálogo de padrões

Classificação segundo **finalidade**:

- **Padrões de criação** - se preocupam com o processo de criação de objetos
- **Padrões de estrutura** - lidam com a composição de classes ou de objetos
- **Padrões de comportamento** - caracterizam a maneira como classes e objetos interagem e distribuem responsabilidades

Catálogo de padrões

Classificação segundo **escopo**:

- **Padrões para classes** - lidam com relacionamentos entre classes e suas subclasses (relacionamentos estáticos)
- **Padrões para objetos** - lidam com relacionamentos entre objetos (relacionamentos dinâmicos)

Catálogo de padrões

	Criação	Estrutura	Comportamento
Classe	Factory Method	Adapter (classe)	Interpreter Template Method
Objeto	Abstract Factory Builder Prototype Singleton	Adapter (objeto) Bridge Composite Decorator Façade Flyweight Proxy	Chain of Responsibility Command Iterator Mediator Memento Observer State Strategy Visitor

Introdução aos padrões de projeto

Exercícios

Exercício

- 1 O que são padrões de projeto de software?
- 2 Quais são os elementos essenciais para descrição de um padrão?
- 3 Os padrões GoF são descritos utilizando quais elementos?
- 4 Quais critérios são adotados na classificação dos padrões de projeto GoF?
- 5 Existem quantos padrões no catálogo GoF?
- 6 Cite um padrão de projeto GoF de criação?
- 7 Cite um padrão de projeto GoF de comportamento?
- 8 Cite um padrão de projeto GoF de estrutura?

Padrões de comportamento

Template Method

Template Method

Comportamento / Classe
(Finalidade) / (Escopo)

Padrões de comportamento

Descrição simplificada do padrão

- A descrição do padrão será simplificada para:
 - Nome e classificação
 - Intenção e objetivo
 - Sinônimos
 - Estrutura
 - Participantes
 - Exemplo de código
 - Consequências

Padrões de comportamento

Template Method

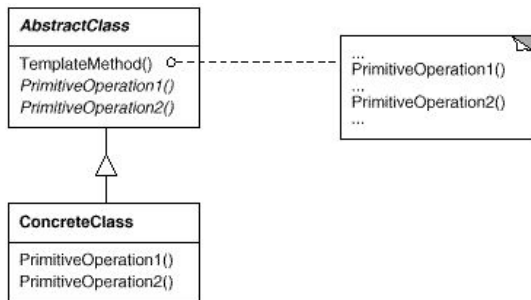
- Intenção e objetivo:

“Definir o esqueleto de um algoritmo em uma operação, postergando alguns passos para subclasses. Permite que subclasses redefinam certos passos de um algoritmo sem mudar a estrutura do mesmo.”

- Sinônimos: “inversão de controle” (princípio de Hollywood)

Padrões de comportamento

Template Method - Estrutura



Estrutura do padrão de projeto Template Method

Obs.: operações primitivas vs. operações gancho

Padrões de comportamento

Template Method - Participantes

- AbstractClass
 - define operações primitivas abstratas que devem ser implementadas nas subclasses
 - implementa um método concreto que invoca operações primitivas abstratas
- ConcreteClass
 - implementa as operações primitivas que executam os passos específicos do algoritmo na subclasse

Padrões de comportamento

Template Method - Exemplo de código

```
public abstract class Conta { // obs. exemplo sem uso de exceptions
    //...

    // template method
    public boolean saque(float valor) {
        if ( verificarestricos(valor) ) {
            realizaSaque(valor);
            return true;
        }
        return false;
    }

    // operacao primitiva 1
    protected abstract boolean verificarestricos(float valor);

    // operacao primitiva 2
    protected abstract void realizaSaque(float valor);

    //...
}
```

Padrões de comportamento

Template Method - Exemplo de código

```
public class ContaCorrente extends Conta {  
    // ...  
  
    protected boolean verificarestricoes(float valor) { // return valor > 0;  
        if (valor > 0) {  
            return true;  
        }  
        return false;  
    }  
  
    protected void realizaSaque(float valor) {  
        if (valor <= saldo()) {  
            setSaldo(saldo() - valor);  
        } else {  
            setSaldo(saldo() - valor - TAXA);  
        }  
    }  
  
    // ...  
  
}
```

Padrões de comportamento

Template Method - Exemplo de código

```
public class ContaPoupanca extends Conta {  
    // ...  
    protected boolean verificarestricoes(float valor) {  
        if (valor > 0 && valor <= saldo()) {  
            return true;  
        }  
        return false;  
    }  
  
    protected void realizaSaque(float valor) {  
        if (valor <= saldo()) {  
            setSaldo(saldo() - valor);  
        }  
    }  
    // ...  
}
```

Padrões de comportamento

Template Method - Consequências

- Benefícios:
 - reuso de código (centralização de partes comuns de algoritmos)
 - comportamento padrão extensível
- Cuidados:
 - é importante diferenciar operações gancho (podem ser redefinidas) de operações abstratas (devem ser redefinidas)

Introdução aos padrões de projeto

Exercícios

Exercício

- 1 Qual a classificação do padrão de projeto Template Method?
- 2 Apresente a estrutura (diagrama de classes) do padrão Template Method?

Introdução aos padrões de projeto

Resumo 1/2

Resumo

- Padrões de projeto são soluções eficientes que já se mostraram úteis em diversos contextos
- Template method - “Definir o esqueleto de um algoritmo em uma operação, postergando alguns passos para subclasses. Permite que subclasses redefinam certos passos de um algoritmo sem mudar a estrutura do mesmo.”

Resumo 2/2

Resumo

- Padrões GoF são descritos através de:
 - Nome e classificação
 - Intenção e objetivo
 - Sinônimos
 - Motivação
 - Aplicabilidade
 - Estrutura
 - Participantes
 - Colaborações
 - Consequências
 - Implementação
 - Exemplo de código
 - Usos conhecidos
 - Padrões relacionados

Leituras recomendadas



Ian Sommerville.

Engenharia de Software

8ª ed., 2007.

Capítulo 18 - Reuso de Software



Sourcemaking Team

Sourcemaking - Teaching IT professionals

<http://sourcemaking.com/>

Perguntas?

Alexandre de Andrade Barbosa
alexandre.barbosa@arapiraca.ufal.br